The Skweezee System: Enabling the Design and the Programming of Squeeze Interactions

Karen Vanderloock[1,2], Vero Vanden Abeele[1,3], Johan A.K. Suykens[2], Luc Geurts[1,2]

[1]e-Media Lab, Group T
Leuven Engineering College
Vesaliusstraat 13
3000 Leuven, Belgium
{karen.vanderloock,
vero.vanden.abeele,
luc.geurts}@groept.be

[2]KU Leuven
ESAT-SCD
and iMinds Future Health
Departement, Leuven, Belgium
Kasteelpark Arenberg 10
3001 Heverlee, Belgium
johan.suykens@esat.kuleuven.be

[3]CUO | Social Spaces
iMinds/KU Leuven
Parkstraat 45
3000 Leuven, Belgium

ABSTRACT
The Skweezee System is an easy, flexible and open system for designing and developing squeeze-based, gestural interactions. It consists of Skweezees, which are soft objects, filled with conductive padding, that can be deformed or squeezed by applying pressure. These objects contain a number of electrodes that are dispersed over the shape. The electrodes sense the shape shifting of the conductive filling by measuring the changing resistance between every possible pair of electrodes. In addition, the Skweezee System contains user-friendly software that allows end-users to define and to record their own squeeze gestures. These gestures are distinguished using a Support Vector Machine (SVM) classifier. In this paper we introduce the concept and the underlying technology of the Skweezee System and we demonstrate the robustness of the SVM based classifier via two experimental user studies. The results of these studies demonstrate accuracies of 81% (8 gestures, user-defined) to 97% (3 gestures, user-defined), with an accuracy of 90% for 7 pre-defined gestures.

Author Keywords
Soft User Interface; Tangible Interaction; Gesture Recognition; Support Vector Machines; Electronic Textiles

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation

INTRODUCTION
Ever since Marc Weiser unleashed his vision of Ubiquitous Computing [31] as computing "weaved into the fabric of our everyday life", this has encouraged HCI researchers to explore how to enrich every day, mundane objects with computation. In particular, tangible interaction has focused on augmenting computing systems with graspable objects [6,29], that can be manipulated in a similar way as non-computing objects, e.g. lifting, rotating or relocating objects on interactive surfaces. Via embedding a myriad of sensors, via the addition of advanced signal processing and computing, ubiquitous computing becomes 'embodied' or part of our everyday interaction [5,10]. In addition to the classical tangible tabletops enriched with graspable widgets [e.g. 12,28], in the past years, a high number of research projects embedded computation in the objects themselves, not relying on larger surfaces [e.g. 4,21,23].

In the past decade, we also witnessed the introduction of electronic textiles in human-computer interaction [1,2]. The goal is to create computing technologies that are entirely fabric based, and can be worn, washed, dried and folded as normal fabrics. E-textiles have been embraced by the wider "do-it-yourself" (DIY) community, not in the least by publications by Buechley and the workshops/website by Perner-Wilson et al. [2,18,20], who set forth the goal to make a library of materials and techniques available that can form the foundation of DIY electronic textiles.

With the Skweezee System, we combine the idea of making interactive devices of everyday objects with the technology and aspiration of electronic textiles. The System contains both a tangible object and an algorithm. The Skweezees are soft tangible objects, filled with conductive padding. When squeezing (by applying pressure with hands or other body parts) the Skweezee is deformed. In order to detect and discriminate specific deformations, the Skweezee contains (soft) electrodes that are dispersed over the shape (eight electrodes in our prototypes). By measuring the resistance between any pair of electrodes, a number is obtained that is related to the magnitude of deformation between those electrodes. For every deformation, a different pattern of measurements is obtained, which allows the computer to distinguish different deformations. Figure 1 shows several potential candidates for Skweezees, ranging from small objects held and squeezed in one hand, to shapes inviting bimanual interaction or even full-body interaction. In essence, any 'soft' object is a candidate for becoming a Skweezee, by replacing the original stuffing with conductive filling, and inserting the desired number of electrical wires which function as electrodes.

UIST'13, October 8–11, 2013, St. Andrews, United Kingdom.
Copyright © 2013 ACM 978-1-4503-2268-3/13/10…$15.00.
http://dx.doi.org/10.1145/2501988.2502033

The Skweezee System also includes software which comprises an algorithm that allows to recognize the squeezing gestures and consists of a Support Vector Machine based classifier. With this software, users can record their own deformations or squeezing gestures that have to be recognized. A graphical user interface shows the aforementioned pattern of measurements resulting from the deformation, and the recognized gesture (see Figure 4).

In summary, the Skweezee System enables end-users to design and develop gestural squeeze interactions at any size and configuration (see Figure 1 and Figure 2). It is our aspiration that people without engineering backgrounds (e.g. therapists, instructors, industrial designers or artists) can design and program their own Skweezees to sense those squeeze gestures that are desired for their application domains. Therefore, Skweezees are very much designed with the same aspiration Buechley et al. [19] put forward: to emphasize end-users' creativity, empowerment and self-expression. In this paper, we will specifically focus on the technology underlying the Skweezee System and test its robustness and accuracy.

RELATED WORK

Soft, deformable objects that act as input devices have been suggested before by other researchers. Perhaps, the most simple form of a Skweezee is the felt pressure sensor introduced by Perner-Wilson et al. [18]. The felt pressure

Figure 1. Potential candidates for Skweezees are any objects that are 'soft', and can be filled with conductive padding and electrodes, rendering the application space infinite, e.g. a cuddly toy or baby toy that can be used for training bimanual skills, a feathered jacket that detects physical caresses, a gym roll or mat for rehabilitation purposes, a stress ball, a pregnancy cushion or neck pillow to assess sleeping patterns or big soft blocks to be used for fun and educational purposes.

Figure 2. Four prototypes of Skweezees: the sphere, the cube, the cylinder and the cuboid.

sensor contains a blend of regular and resistive yarn. When squeezed, the conductive fibers throughout the sensor improve the electrical connections, lowering the resistance between any two electrodes on the sensor's surface. Skweezees differ from felt pressure sensors: they contain more than two electrodes (eight in the prototypes presented in this paper), and measure the resistance between all possible pairs of electrodes. This way a Skweezee can be considered an N-dimensional pressure sensor ($N>1$), allowing to detect more complex deformations of the sensor.

Murakami and Naomasa also investigated soft input devices, and more specifically 3-D deformable shapes [17]. They constructed deformable objects consisting of up to ninety small bars of conductive polyurethane foam. When deformed, the length of several bars will change and thus their resistance as well. By deriving the lengths of all bars from their resistance, the geometric shape deformation of the object can be estimated. Smith et al. used conductive foam as well for their Digital Foam, described as "a new input sensor developed to support clay like sculpting and modeling operations" [26]. Their prototype consists of 162 discrete bars of conductive foam embedded in a sphere, each acting as a unique pressure sensor. Skweezees have a somewhat simpler construction, consisting of a textile covering filled with homogeneous conductive fabric. Skweezees do not aim to determine their exact geometric deformed shape, but rather the gesture causing the deformation, as performed and recorded earlier by the user.

Slyper et al. suggested silicone as base material for their soft sensor [25]. While silicone provides the necessary mechanical properties (deformable and elastic), electrical (non-deformable) switches are placed at several places in the object, and are opened or closed depending on the user's action. The materials used in Skweezees do not limit the deformation, there are no rigid, non-deformable sensors inserted. Moreover, sensing in Skweezees is continuous, as opposed to the discrete nature of the measurements in Slyper's work.

Sensing deformation through measuring changes in conductivity has also been done in the context of bendable interfaces like, for example TWEND [8] or PaperPhone [15]. TWEND consists of two pieces of thin, flexible plastic with a layer of foam in between, in which eight optical bending sensors are embedded. PaperPhone consists of a flexible display, augmented with a layer of bidirectional bend sensors. Both, however, are not fully twistable and can be damaged due to the characteristics of the used material. Skweezees are not limited in the kind of deformations, and cannot 'break'.

Sensor wise, this work could be seen as an extension of the TactileTape sensor [9] to three-dimensional instead of one-dimensional volumes. TactileTape is a one-dimensional touch sensor that looks and behaves like regular tape. However, it is actually a flexible potentiometer, consisting of three surfaces that form an open circuit. When a finger deflects the surface, the circuit closes.

PinStripe [13] resembles our Skweezee technology in that it also relies on the change of resistance. Pinstripe consists of fields of parallel conductive lines sewn onto the fabric. It can be controlled by varying the amount of cloth pinched. However, resistivity measurements between each pair of lines are only binary. In the Skweezee System, the resistivity measurements allow for a continuous range of values. Moreover, the Skweezee System enables 3-D deformations, rather than the two-dimensional sheet of fabric for the PinStripe.

Finally, Sugiura et al. presented the FuwaFuwa sensor module [27], which measures deformation in six orthogonal directions via photo reflectivity. The sensor is "a round, hand-size, wireless device for measuring the shape deformations of soft objects such as cushions and plush toys." To do so, this sensor (a ball with a diameter of 65mm) has to be placed in the soft deformable object. In case more distant deformations have to be measured, the authors suggest to insert another module. In the case of a Skweezee, the object to be deformed *is* the sensor, no extra modules need to be inserted, and the object can take any size or geometric shape.

TECHNOLOGY OF SKWEEZEES

In order to test the feasibility of Skweezees, we made several prototypes intended to be deformed with two hands (see Figure 2). We explored several shapes, materials, electrode fixtures and electrode positions.

Materials

As mentioned before, Skweezees are filled with conductive padding. The Skweezee should be deformable to a large extent (the order of magnitude of deformation runs in centimeters for our prototypes) and return to its rest position when untouched. Therefore, the filling should show considerable elasticity. The filling should be conductive as well; its resistance should drop when pressed together. We found the balance between conductivity and elasticity to be rather delicate. We experimented with nylon filaments suffused with conductive carbon (Resistat F9116, www.resistat.com), conductive wool consisting of steel fibers mixed with normal wool (Bekinox W12/18, www.bekaert.com) and low-density conductive foam. Our experiences gave us a preference for conductive wool, but obviously, one can experiment with other fillings as well, as long as one bears in mind the need for both elasticity and conductivity.

Figure 3. Inside view of the textile casing of a Skweezee, showing the electrode consisting of conductive wire, conductive tape and a sample of the filling, all stitched together with conductive yarn.

Inside the Skweezees, we embedded electrodes, dispersed over the object. In our shapes we embedded eight, but embedding more or less electrodes is possible. The number should be defined by the number of gestures one aims to recognize, by the size and shape of the object, and the targeted accuracy of gesture detection. We will revisit this choice of number of electrodes in the Discussion section.

Different options are possible to fixate the electrodes insight the shape. It is critical that electrodes will not migrate when the shape is deformed and that they have a permanent contact with the conductive padding. Therefore, one should look for a mechanism to keep the electrode in its place and in permanent contact with the filling (see Figure 3). We relied on using conductive tape to fixate the electrode to the outer lining of the shape. To ensure its position, the electrodes were not only taped but also secured by a double stitch with conductive yarn.

Electronics

When deforming a Skweezee the resistance between two or more electrodes will change. We developed a circuit to measure the resistance between every unique pair of electrodes. Our prototypes contain eight electrodes (N=8), so there are 28 (=N*(N-1)/2) unique electrode pairs to be scanned. Two multiplexers – one for each electrode of the pair – are used to select an electrode pair. The multiplexers are driven by the digital outputs of a microcontroller (Arduino UNO). A voltage divider and the microcontroller's A/D converter is used to measure the resistance between the two selected electrodes. All 28 measurements are then sent to the PC.

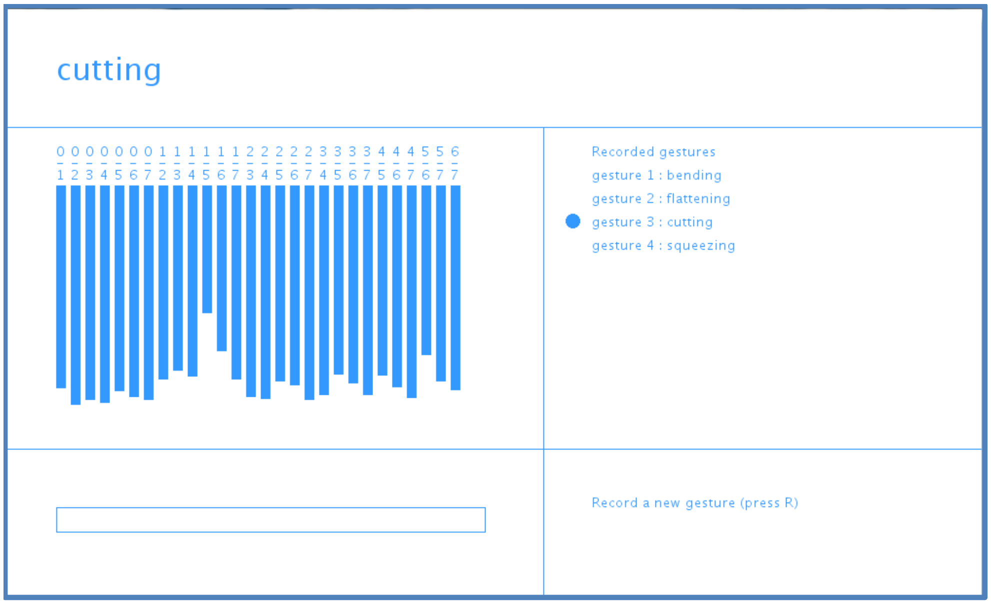

Figure 4. Screenshot of the software showing the 28 measurements (bars at the left) and the names of the recorded gesture (at the right). The recognized gesture is shown at the top.

Software

Software for recording and sensing gestures is written in Processing. The software allows the user to monitor all measurements obtained when deforming the Skweezee, in real time (see the 28 bars on the left side, on the screenshot in Figure 4). The program shows the names of all user-defined gestures (i.e. shape deformations) (see the right side of the screenshot in Figure 4). The name of the recorded gesture that is the most similar to the current one when performing a gesture, is shown at the top of the screen, (see "cutting" on the screenshot in Figure 4).

The user can also record additional gestures. First, s/he gives a name to the gesture, then s/he changes the shape of the Skweezee. During a time interval of 10 seconds, the user can perform the gesture once, or several times, and dynamically change the extent of the deformation. All instances for which the sample values exceed a preset threshold (defined in a pilot test) are stored in memory, and serve as a reference for the gesture.

Gesture recognition algorithm

Every gesture generates different consecutive patterns of 28 measurements that correspond to the measurements across the 28 unique electrode pairs. Some examples of such patterns at a certain moment in time are shown in Figure 5. A classifier algorithm is then needed to discern these patterns. When a new gesture is performed, the classifier has to decide which patterns in the recorded set are most similar to the new gesture. The first implementation was a minimum-distance classifier. However, the performance of this classifier was unsatisfactory, since it could not detect similar gestures with different amplitudes. As a result, the user had to perform the gesture always to the same extent.

To solve this, we opted for a Support Vector Machine (SVM) classifier [24,30]. The features on which the classifier bases its decision are the 28 measurements, recorded continuously from the moment the user starts the deformation until he releases the Skweezee. Each recorded instance corresponds to a single point x_i in a 28-dimensional space. During the recording of a gesture, the data of the 28 measurements are continuously saved, so multiple points x_i are obtained. All these different instances are labeled with the correct gesture y_i. This list of instances are then added to the previous recordings, and so a new training set $\{x_i, y_i\}_{i=1}^{N}$ is obtained. Typically, around 110-170 points are obtained for each gesture. The basic concept of SVMs is the use of a hyper plane to separate data of two classes. To extend the classifier to multiple classes the "one-against-one" method with "max wins" voting is used, based on the comparison in [11].

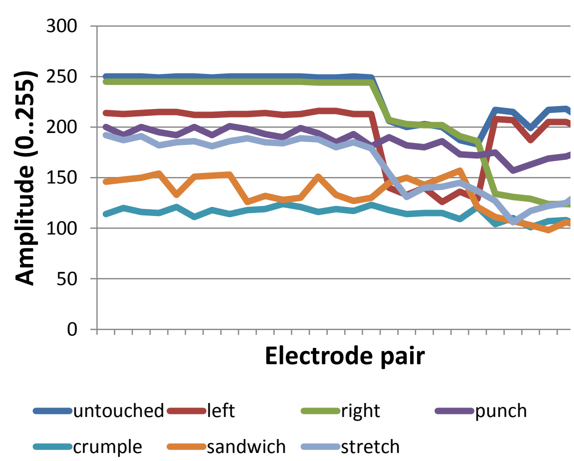

Figure 5. Pattern of measurements corresponding to the gestures shown in Figure 6.

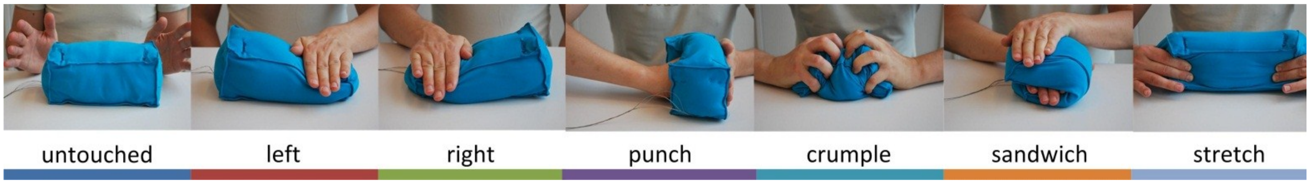

Figure 6. The seven predefined gestures with the cuboid.

The idea behind an SVM classifier is to map the original data points to a high-dimensional, or even infinite-dimensional, feature space so the classification problem becomes easier. The mapping φ is done by an appropriate choice of a kernel function, $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$.

Based on [14] and [16], the radial basis function (RBF) is chosen:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0.$$

SVMs require the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$$

Subject to

$$y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i, i = 1, \dots, N,$$

$$\xi_i \geq 0, i = 1, \dots, N.$$

$C > 0$ is the penalty parameter of the error term. The library LIBSVM [3] is used to train the classifier and predict new values. To be able to use the default values for C and γ, each of the 28 measurements of the training set is linearly scaled to the range $[-1, +1]$ using the formula:

$$x_{i,scal}(l) = -1 + 2 * \frac{x_i(l) - \min(feature\ l)}{\max(feature\ l) - \min(feature\ l)},$$

$$l = 1, \dots, 28\ and\ i = 1, \dots, N$$

where $x_i(l)$ is the l^{th} element of the 28-dimensional vector x_i and $x_{i,scal}(l)$ is the scaled measurement l. The scaling factors, i.e. the minimum and maximum of each feature of the training data, are saved. The same scaling for each measurement is then applied on each instance of a performed gesture during the test phase.

The best parameters C and γ can be found by doing a "grid-search" using cross-validation. We want to avoid to do this model selection each time a new gesture is recorded, because we want to keep the recording of a new gesture real-time. As a result, we have done a grid-search for C and γ offline, on different data sets. Of course, the optimal parameters change depending on the gestures that are recorded. Following parameters are chosen: $C = 1$ and $\gamma = 0.07$. For this choice, we got offline cross-validations between 89.2% and 100%, depending on the data set.

Finally, we want to stress that the complexity of this SVM classifier is hidden for the user. Users can simply perform and record their gestures in the software, and the classification happens automatically.

EVALUATION OF SKWEEZEES

In order to assess the feasibility of the Skweezee System and more particularly the accuracy of the classifier, we conducted two experiments with two of our prototypes, namely the cylinder and the cuboid (see Figure 2). The cuboid's measurements are 10 cm x 10 cm x 20 cm. The cuboid contains 8 electrodes, each positioned in one of the corners. The cylinder has a diameter of 7 cm and a length of 22 cm. One electrode is positioned at each end, two pairs of three electrodes are equally distributed around the circumference at one third and two third of the length respectively.

Participants

For the first experiment, we invited 20 participants, 14 male and 6 female, with an age between 19 and 59. For the second experiment, we invited 10 other participants, 7 male and 3 female, aged between 19 and 52. None of them had experimented with our Skweezee System before, they were unaware of the characteristics of the Skweezees at hand.

Experiment 1: Pre-defined gesture set

Methods and rationale

The first experiment lasted on average 10 minutes, and consisted of a single test run for one of the two Skweezees. 10 participants tested the cuboid, 10 other participants tested the cylinder. For each Skweezee, a set of gestures was defined by the experimenter before the experiments were carried out. The predefined set consisted of seven gestures. Figure 6 shows the gestures for the cuboid: "untouched", "left", "right", "punch", "crumple",

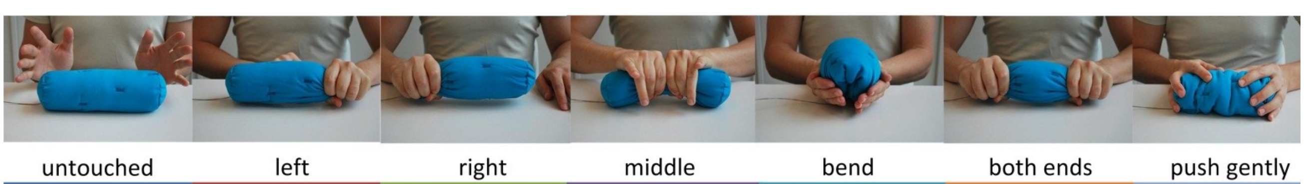

Figure 7. The seven predefined gestures with the cylinder.

"sandwich" and "stretch". The gestures for the cylinder are shown in Figure 7: "untouched", "middle", "bend", "both ends", "push gently", "left" and "right". These gestures were selected and recorded by the experimenter, based on his knowledge of the characteristics of each Skweezee (e.g. the size, the shape and the position of the electrodes). The recorded data were included in the training set of the classifier.

For the experimenter, it was always possible to perform each gestures in such a way that the computer recognized it correctly, an accuracy of 100% was achieved. However, we wanted to find out whether the classifier still performed well when the gestures were performed by other users. Since it was expected that every participant would make the gestures in a slightly different way (size and positioning of the hands, applied force when squeezing,…), it was likely that the accuracy of the classifier would drop (as shown in similar studies, e.g. [22]). The data 'generated' by the participant are considered as the test set of the classifier (and obviously different from the training set).

Procedure
At the start of the experiment, the experimenter demonstrated all gestures, and then asked the participants to make the same gestures with the Skweezee. During this practice phase, the participant was allowed to look at the computer screen to see which gesture was recognized by the classifier. Next, the evaluation phase was carried out, during which no visual or auditory feedback was given to the participant. The researcher called out the name of one of the seven gestures in the set, and then the participant had to perform the according gesture within three seconds, after which the recognized gesture was logged. The gestures were called out randomly, and the researcher ensured that each gesture was called out four times. Hence, after completing, the participant had performed 28 gestures.

Results
On average, a classification accuracy of 90% was obtained for both Skweezees, with a standard deviation of 8%. Two participants succeeded to get a score of 100%. The lowest score measured is 71%.

Experiment 2: User-defined gesture set

Methods and rationale
The motive for the second experiment was to evaluate the Skweezees when users were allowed to create their own gesture set without a specific task in mind. Contrary to other studies [7,22,25], we did not tell the participants which gestures they had to perform. Neither did we inform the participants about the potential and the limitations of the Skweezees, which might inspire (or constrain) the creation of an optimal gesture set. As shown in [32], users can come up with other gestures than the experts and vice versa. However, if the performance of the system would still be acceptable, Skweezees would open up a lot of opportunities for people wishing to use Skweezees for their own applications.

Procedure
The experiment lasted on average 40 minutes, and consisted of three test runs for each of the two Skweezees. For each test run, first there was a recording phase, to generate the training set of the classifier, and next an evaluation phase, to generate test samples for the classifier. During the recording phase, the participant was asked to think of a squeeze gesture, and a corresponding name. When ready, the researcher would type in the name and the participant was asked to perform the actual gesture in order to record it. The recording lasts 10 seconds, so the participant was asked to perform the same gesture a few times, while releasing and regrabbing the Skweezee, to be able to catch different intensities and different ways of grabbing. When four different gestures were programmed, the classifier was automatically trained and we switched to the evaluation phase. During the evaluation phase, the researcher would

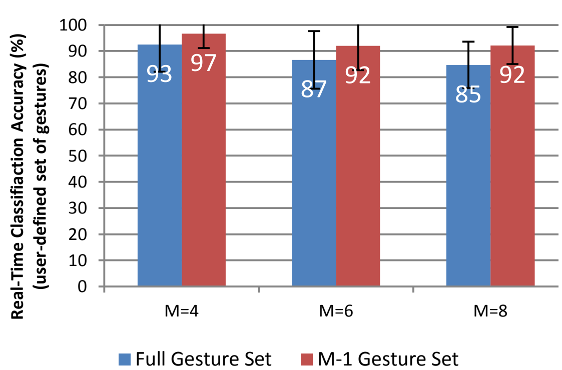

Figure 8. Real-time, per-user classification accuracy for 4, 6 or 8 gestures on the cuboid (in blue). When the 'worst' gesture is removed, scores improved (in red). The error bars represent the standard deviations of these accuracies.

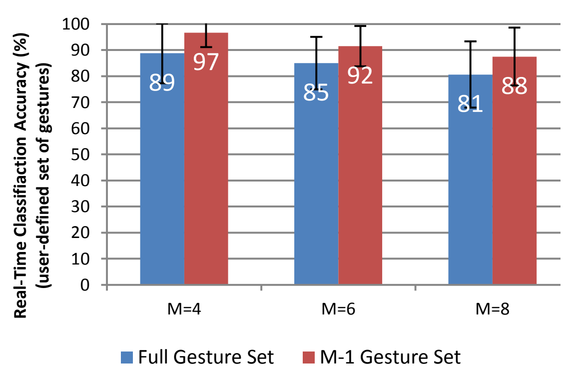

Figure 9. Real-time, per-user classification accuracy for 4, 6 or 8 gestures on the cylinder (in blue). When the 'worst' gesture is removed, scores improved (in red). The error bars represent the standard deviations of these accuracies.

call out the name of one of these four gestures and ask the participant to perform the according gesture. The gestures were called out randomly, and the researcher ensured that each gesture was called out four times. No feedback was given during the evaluation phase, so the participant could not make corrections if an incorrect gesture was recognized. The participant had to complete the gesture within three seconds. Hence, after completing the first trial, the participant performed 16 gestures. In the next test run, two more gestures were added according to the same procedure, hence totaling six gestures. Finally, the third trial consisted of eight gestures. We decided not to continue and add more gestures, since most participants indicated that it was difficult to remember more gestures and to invent new ones. The same procedure was adopted for both Skweezees.

Results

Figure 8 (for the cuboid) and Figure 9 (for the cylinder) show the real-time classification accuracies, averaged across all participants, together with the standard deviations shown as error bars on the figures. As can be inferred from the graphs (in blue), for the cuboid we achieved accuracies between 85% and 93% for the full gesture set (ranging from 4 to 8 gestures). The individual maximum score for 4 and 6 gestures was 100%, and 97% for 8 gestures. For the cylinder, accuracies between 81% and 89% are obtained. For each size of gesture set, an individual accuracy of 100% could be obtained. When the 'worst' gesture is removed, i.e. the results of the gesture with the lowest recognition rate is not included, the scores for the cuboid vary between 92% (7 gestures) and 97% (3 gestures). For the cylinder, these numbers range from 88% to 97%. Removing the worst score is also done in other similar experiments, e.g. [22], and somewhat compensates for the fact that the participants were inexperienced, as discussed below.

DISCUSSION

Accuracy

Our experimental evaluation demonstrates the feasibility of Skweezees, achieving real-time accuracies between 81% and 97%. One needs to keep in mind that these accuracies are obtained by only one recording interval of 10 seconds. From a user perspective, this is a great benefit: the user only has to record each gesture once. However, we expect the recognition rate to increase when taking more samples per gesture type in the user-defined case and data from more participants in the pre-defined scenario.

In addition, we like to stress again that participants had no prior experience with the Skweezees. They did not receive information on how the Skweezees were designed (e.g. where the electrodes were located in the shape and where the greatest resistance changes would be). Henceforth, participants did not have the knowledge on how to create an 'optimal' set of gestures that could easily be discriminated by the Skweezee. The programmed gestures in their set were the result of the participants' own creativity and the perceived and real affordances of the Skweezee only.

Another remarkable observation is the big difference between participants. Especially during the second experiment, some participants really tried to find gestures with a unique pattern and really saw it as a challenge to define different gestures. They enjoyed creating the different gestures. As an example, with 8 user-defined gestures, one participant scored 97% for the cuboid and 100% for the cylinder. We also noted that during the second experiment, some users did find it hard to come up with eight different 'meaningful' gestures. Moreover, users found it hard to remember and faithfully reproduce them later on. Hence, we subsume that user-defined limitations on the amount of gestures that can be memorized and reproduced might have affected our accuracies in a negative way.

Advantages

The Skweezee System has some important advantages over existing technologies for soft interfaces. First, the Skweezee System has a simple construction, consisting of a textile covering filled with homogeneous conductive fabric and some electrodes. The object to be deformed is the sensor. Secondly, they can detect a wide range of variations in deformation. Moreover, the Skweezee System does not determine the exact geometric deformed shape, but rather the gesture causing the deformation, as performed and recorded earlier by the user. As shown in the experimental results, the Skweezee system achieves accuracies between 81% and 97%. Finally, the Skweezee System is easy to use: it can be designed and programmed by those active in the 'field' themselves, coaches, therapists, instructors, etc. We can think of cuddly toys in physical therapy that encourage bimanual actions, input devices that facilitate intuitive 3D drawing or 3D manipulation, or even outdoor furniture that stimulates exergaming. But certainly, end-users can envision even more creative and embodied designs.

Limitations

The most important limitation of the current system is its 'sensitivity'. Sometimes, small changes in the deformation lead to relatively large differences in the obtained pattern of measurements. On the other hand, totally different gestures can result in a quite similar pattern of measurements. This paradoxical observation can partially be explained by the limited resolution of the system. Indeed, a decrease of resistance between two electrodes is due to compressed material in between them, but the system cannot tell where exactly this compression is (close to one of both electrodes, in the middle, equally distributed,…). On the other hand, changes in the intensity of the deformation are easily detected, which explains the high sensitivity to small changes. One way to solve these issues is by increasing the number of electrodes, which in turn will increase the resolution of the Skweezee, and so the accuracy of gesture classification. The underlying technology does not specify that only eight electrodes need to be used. However, there is a trade-off with complexity of the design and with the speed of measurement. With every extra electrode, the number of measurements and computation time increase as well. We have not experimented with more electrodes, therefore we can neither specify the influence on accuracy nor the influence on computational time. Nevertheless, whatever the results of adding electrodes, accuracies will also be defined by the design of the shape (inviting specific gestures through its shape) and the according positioning of the electrodes. Investigating the effect of more electrodes and a higher resolution is part of our future work.

We hope that given the generic nature of the Skweezee System, even novices can easily create a shape that incorporates desired affordances, position electrodes where the measurement would make the most difference, and program the desired squeeze gestures. Further work will include Skweezee workshops with non-experts, to investigate whether designs of Skweezees and the recording of gestures is as straightforward as we hope for.

Another limitation which is not further investigated is the lifetime of the Skweezees. We assume that the conductive wool compresses over time, which will affect the sensor data. Also, the proposed platform only works for soft objects that have no other functions since the inner layer is restricted with the material needed for the sensing.

CONCLUSION

Skweezees are soft, squeezable shapes that are filled with conductive padding and strategically positioned electrodes. Using a Support Vector Machine classifier, the Skweezee System can be programmed, even by non-techies, to sense their shape deformation. Consequently, the Skweezee System enables rich gestural squeeze interaction for the DIY community. Our experimental results show that well designed and crafted Skweezees can easily sense up to 7 different gestures with an accuracy of 90%, without extensive training. Hence, we are convinced that the Skweezee System can become an additional tool to bring ubiquitous computing to the "real world" and help users in their quest for creativity, empowerment and self-expression.

ACKNOWLEDGMENTS

This work is part of the TIII project, a TETRA funded research project by IWT – Flanders (120147). Johan Suykens acknowledges support from KU Leuven, the Flemish government, FWO, the Belgian federal science policy office and the European Research Council (CoE EF/05/006, GOA MANET, IUAP DYSCO, FWO G.0377.12, ERC AdG A-DATADRIVE-B).

REFERENCES

1. Berzowska, J. Programming materiality. *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, ACM (2012), 23–24.
2. Buechley, L. and Eisenberg, M. Fabric PCBs, electronic sequins, and socket buttons: techniques for e-textile craft. *Personal and Ubiquitous Computing 13*, 2 (2007), 133–150.
3. Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. 2*, 3 (2011), 27:1–27:27.
4. Collective, B.M. and Shaw, D. Makey Makey: improvising tangible and nature-based user interfaces. *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, ACM (2012), 367–370.
5. Dourish, P. *Where the action is: the foundations of embodied interaction*. MIT Press, Cambridge, MA, USA, 2001.
6. Fitzmaurice, G.W., Ishii, H., and Buxton, W.A.S. Bricks: laying the foundations for graspable user interfaces. ACM Press/Addison-Wesley Publishing Co. (1995), 442–449.
7. Geurts, L., Van Woensel, A., and Vanden Abeele, V. No sweat, no fun: Large-gesture recognition for computer games. *Proceedings of the fourth international conference Fun and Games 2012*, (2012).
8. Herkenrath, G., Karrer, T., and Borchers, J. Twend: twisting and bending as new interaction gesture in mobile devices. *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, ACM (2008), 3819–3824.
9. Holman, D. and Vertegaal, R. TactileTape: low-cost touch sensing on curved surfaces. *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, ACM (2011), 17–18.
10. Hornecker, E. and Buur, J. Getting a grip on tangible interaction: a framework on physical space and social interaction. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM (2006), 437–446.
11. Hsu, C.-W. and Lin, C.-J. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks 13*, 2 (Mar), 415–425.
12. Jordà, S. The reactable: tangible and tabletop music performance. *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, ACM (2010), 2989–2994.
13. Karrer, T., Wittenhagen, M., Lichtschlag, L., Heller, F., and Borchers, J. Pinstripe: eyes-free continuous input on interactive clothing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2011), 1313–1322.
14. Keerthi, S.S. and Lin, C.-J. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural computation 15*, 7 (2003), 1667–1689.
15. Lahey, B., Girouard, A., Burleson, W., and Vertegaal, R. PaperPhone: understanding the use of bend gestures in mobile devices with flexible electronic paper displays. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2011), 1303–1312.
16. Lin, H. and Lin, C.-J. *A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods*. 2003.
17. Murakami, T. and Nakajima, N. Direct and intuitive input device for 3-D shape deformation. *Proceedings of*

the SIGCHI conference on Human factors in computing systems: celebrating interdependence, ACM (1994), 465–470.

18. Perner-Wilson, H., Buechley, L., and Satomi, M. Handcrafting textile interfaces from a kit-of-no-parts. *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, ACM Press (2011), 61.
19. Perner-Wilson, H. and Buechley, L. Making textile sensors from scratch. *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, ACM (2010), 349–352.
20. Perner-Wilson, H. and Satomi, M. How to get what you want. *KOBAKANT DIY Wearable Technology Documentation*. http://www.kobakant.at/DIY/.
21. Ryokai, K., Marti, S., and Ishii, H. I/O brush: drawing with everyday objects as ink. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2004), 303–310.
22. Sato, M., Poupyrev, I., and Harrison, C. Touche;: enhancing touch interaction on humans, screens, liquids, and everyday objects. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, ACM (2012), 483–492.
23. Schiettecatte, B. and Vanderdonckt, J. AudioCubes: a distributed cube tangible interface based on interaction range for sound design. *Proceedings of the 2nd international conference on Tangible and embedded interaction*, ACM (2008), 3–10.
24. Schölkopf, B.S. and Smola, A.J. *Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.
25. Slyper, R., Poupyrev, I., and Hodgins, J. Sensing through structure: designing soft silicone sensors. *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, ACM (2011), 213–220.
26. Smith, R.T., Thomas, B.H., and Piekarski, W. Digital foam interaction techniques for 3D modeling. *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, ACM (2008), 61–68.
27. Sugiura, Y., Kakehi, G., Withana, A., et al. Detecting shape deformation of soft objects using directional photoreflectivity measurement. *Proceedings of the 24th annual ACM symposium on User interface software and technology*, ACM (2011), 509–516.
28. Ullmer, B. and Ishii, H. The metaDESK: models and prototypes for tangible user interfaces. *Proceedings of the 10th annual ACM symposium on User interface software and technology*, ACM (1997), 223–232.
29. Ullmer, B. and Ishii, H. Emerging frameworks for tangible user interfaces. *IBM Syst. J. 39*, 3-4 (2000), 915–931.
30. Vapnik, V.N. *Statistical learning theory*. Wiley, 1998.
31. Weiser, M. The Computer for the Twenty-First Century. *Scientific American 265*, 3 (1991), 94–104.
32. Wobbrock, J.O., Morris, M.R., and Wilson, A.D. User-defined gestures for surface computing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2009), 1083–1092.